

Graphic Applications Development

ITAcademy

Predavač



Dušan Čoko

master inženjer elektrotehnike i računarstva strukovnih studija

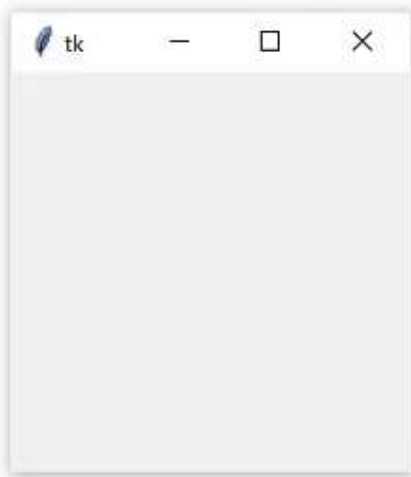
dusan.coko@it-akademija.com

- Profesionalno se bavi web programiranjem od 2011.
- Profesionalno se bavi informacionim sistemima od 2013.
- Saradnik u nastavi na Visokoj školi elektrotehnike i računarstva od 2015 do 2019.
- Od 2018. honorarno angažovan kao predavač na IT Akademiji
- Od 2021. tehnički direktor u Allios, oblast hardversko softverskih rešenja za telekomunikacije, industrijsko inženjerstvo i bezbednost.

Tkinter

- Tkinter biblioteka(“Tk interface”) je standardni Python interfejs za TK GUI toolkit. Tk GUI toolkit je dostupan na većini Unix platformi (Linux/Mac OS) kao i Windows sistemima.
- Sam TK GUI toolkit nije deo Python-a već je grafička biblioteka kojoj tkinter pristupa. Posledica ovoga je da su moguća određena vizuelna odstupanja kod prikaza grafičkih elemenata.
- Aplikacija napisane kroz Tkinter garantuje da će koristiti grafičke elemente operativnog sistema tako da aplikacije bi trebalo da funkcionišu nezavisno od platforme.

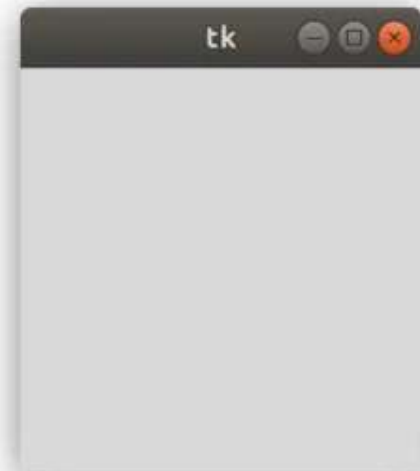
Tkinter



(a) Windows



(b) macOS



(c) Ubuntu

ITAcademy

Tkinter

- Za uvoz grafičkih komponenti u program koristi se biblioteka ***tkinter***.
- Glavni element grafičke aplikacije je ***Window*** element. Window element se kreira korišćenjem klase ***Tk***.
- Da bi prozor ostao prikazan (a ne odmah se ugasio pri pokretanju), window element ima metodu ***mainloop*** koja drži otvoren prozor i osluškuje sve događaje.

Tkinter

```
from tkinter import *  
  
root = Tk()  
  
if __name__ == "__main__":  
    root.mainloop()
```

Tkinter

- Za podešavanje naslova prozora koristi se metoda ***title***.
- Za podešavanje dimenzija prozora koristi se metoda ***geometry***. Metoda prihvata string u formatu ***widthxheight***. Vrednosti su u pixel-ima.
- Za dodavanje ikonice prozora koristi se metoda ***iconbitmap***. Metodi se navodi relativna ili absolutna putanja do ikonice.

Tkinter

```
from tkinter import *  
  
root = Tk()  
root.title("Hello World")  
root.geometry("400x300")  
root.iconbitmap("./graphics/travel.ico")  
  
if __name__ == "__main__":  
    root.mainloop()
```


Widgets

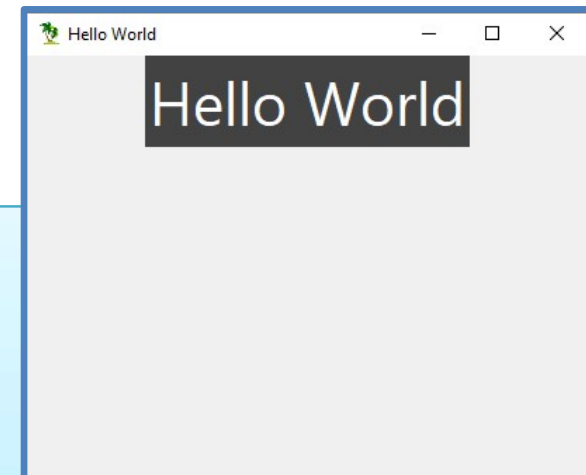
- Tkinter se sastoji od komponenti koje se zovu **widgets**.
- Mogu se koristiti ugrađeni widget-i a mogu se pisati sopstveni.
- Svi widget-i moraju imati svog roditelja (master). Najgornji roditelj je window, a postoje i drugi kontejner elementi kao paneli, radio grupe itd.
- Widget-i se vizuelno prikazuju korišćenjem metode **pack** koja smešta element na roditelja ili korišćenjem metode **grid** koja dozvoljava da se element postavi na mrežu roditelja.
- Takođe korišćenjem metode **place** moguće je definisati koordinate widget-a.

Label

- **Label** je grafički element za prikaz teksta.
- Label koristi argument **text** za inicijalnu postavku teksta.
- Label koristi argument **font** za inicijalnu postavku fonta. Prosleđuju se kao String naziv fonta, i veličina fonta (numerički - pixeli).
- Za boju pozadine iza teksta labele mogu se koristiti svojstva **bg/background** ili **fg/foreground**. Vrednosti su naziv boje ili heksadecimalna vrednost.

Label

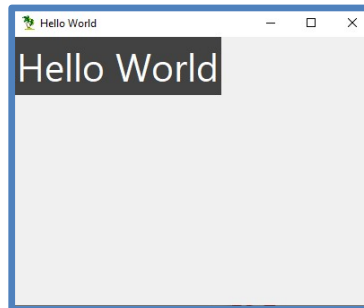
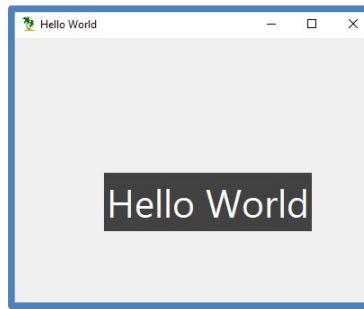
```
from tkinter import *  
  
root = Tk()  
  
...  
label = Label(root, text="Hello World",  
               background="#424242",  
               fg="white", font=("Segoe UI", 32))  
label.pack()  
if __name__ == "__main__":  
    root.mainloop()
```



Label

```
label.place(x=100,y=150  
)
```

```
label.grid(row=0,column=0)
```



Label

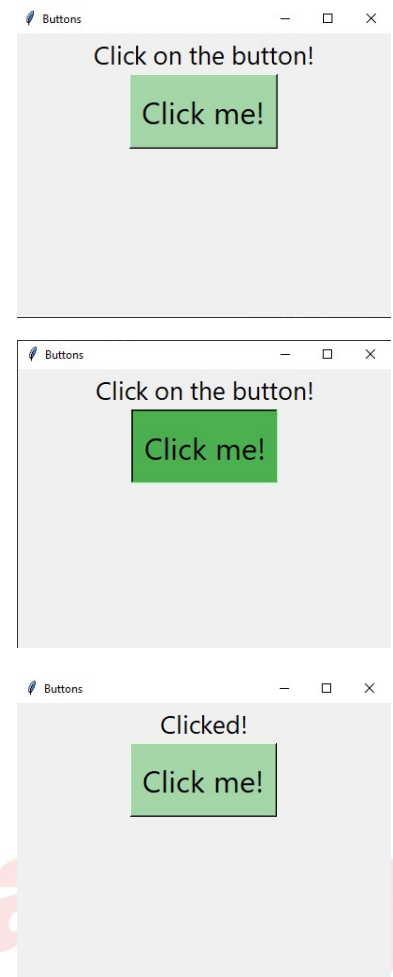
- Sa argumentom ***width*** definiše se broj karaktera (ne pixela) koje labela prikazuje. Ukoliko nije definisano svojstvo labela uzima veličinu njenog sadržaja.
- Sa argumentom ***height*** definiše se vertikalna dimenzija labele u pixelima.
- Sa argumentom ***anchor*** definiše se pozicioniranje teksta ukoliko je širina veća nego sadržaj. Opcije su LEFT, RIGHT i CENTER (podrazumevana vrednost).
- Sa argumentom ***wraplength*** labelu je moguće razvojiti u više redova kada jedan red pređe ovaj broj karaktera.
- Svojstvima label-a i drugih widget-a pristupa se kao rečniku pa je moguće promeniti svojstva kao ***objekat[svojstvo]=vrednost***.
- Drugi način je korišćenjem objekta ***StringVar*** koje se labeli dodeljuje kroz svojstvo ***textvariable***. Kada se promeni vrednost objekta menja se i prikaz u labeli.

Button

- Button je ugrađeni widget koji oponaša različite vrste dugmadi.
- Button koristi argument **command** koje vezuje funkciju ili metodu za klik aktivnost dugmeta.
- Button koristi argument **activebackground** da definiše boju pozadine kada je dugme kliknuto.
- Button koristi argument **activeforeground** da definiše boju fonta kada je dugme kliknuto.
- Button koristi argument **relief** da definiše tip okvira. Neke od vrednosti su **SUNKEN**, **RAISED**, **GROOVE** i **RIDGE**.
- Sa argumentom **state** definišemo stanje dugmeta. Podrazumevano je **NORMAL**, **ACTIVE** je kada je miš na dugmetu i **DISABLED** je ukoliko želimo da isključimo dugme.

Button

```
...  
LABEL_TXT = StringVar()  
LABEL_TXT.set("Click on the button!")  
myLabel = Label(root,  
                 textvariable=LABEL_TXT,  
                 font=("Segoe UI", 20))  
  
def change_text():  
    global LABEL_TXT  
    LABEL_TXT.set("Clicked!")  
  
myButton = Button(root, text="Click me!",  
                  command=change_text,  
                  font=("Segoe UI", 24),  
                  background="#a5d6a7",  
                  activebackground="#4caf50")  
  
myLabel.pack()  
myButton.pack()
```



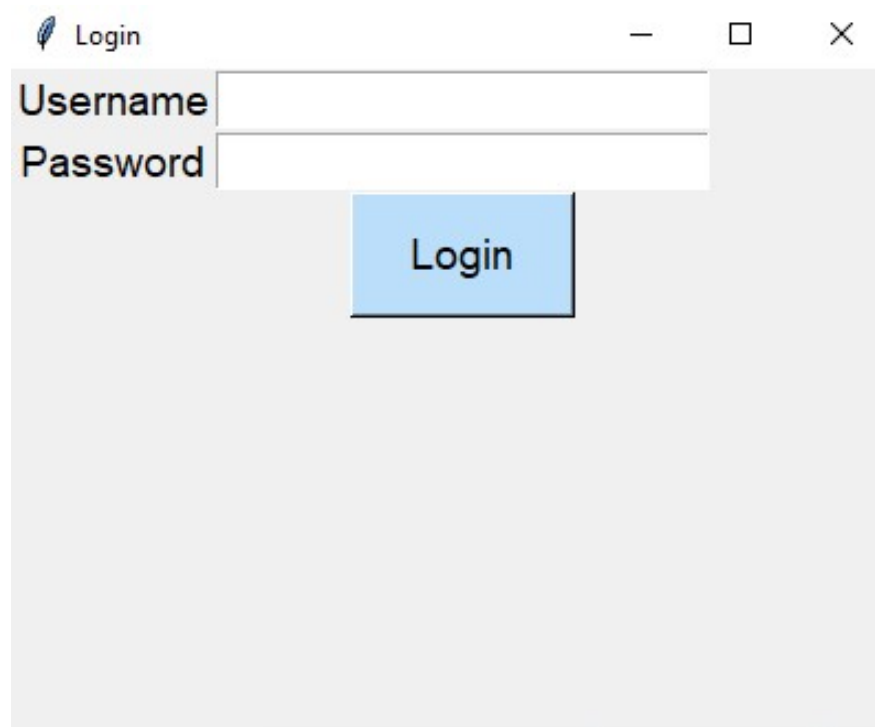
Entry

- Entry je widget za unos jedne linije teksta.
- Entry koristi argument ***show*** da karaktere sakrije korišćenjem nekog drugog karaktera (npr. * za lozinke).
- Za uzimanje vrednosti koristi se metoda ***get***.
- Za brisanje unosa koristi se metoda ***delete***.
- Za dodavanje teksta koristi se metoda ***insert***. Prvi argument je indeks na kom se dodaje tekst a drugi vrednost. Postojeći tekst se neće menjati.
- Preporuka je da se koristi StringVar za manipulaciju vrednosti.

ITAcademy

Primer

- Kreirati interfejs kao na slici.
- Kreirati događaj na klik dugmeta koji proverava da li su podaci uneti za username i password jednaki hardkodovanim podacima “admin” i “1234”.



The image shows a mockup of a login window. It has a title bar with a feather icon, the text 'Login', and standard window controls (minimize, maximize, close). Below the title bar, there are two labels: 'Username' and 'Password'. Each label is followed by a text input field. To the right of these fields is a blue button with the text 'Login'.

Check Button

- Check button je widget koji može da ima jedno od dva stanja - čekirano i nečekirano.
- Sa argumentom **text** definiše se tekst koji stoji uz dugme.
- Sa argumentom **onvalue** definiše se vrednost kada je dugme čekirano, a sa argumentom **offvalue** kada nije.
- Sa argumentom **variable** prati se vrednost stanja checkbox-a. Za snimanje stanja koristi se objekat **IntVar**.

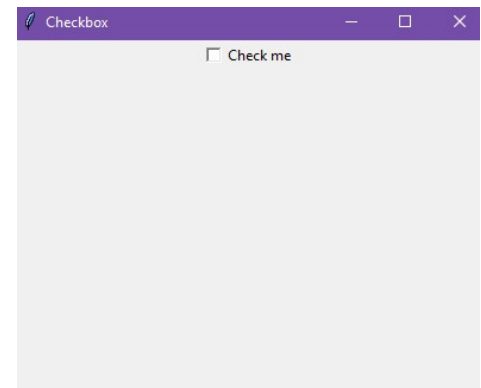
ITAcademy

Check Button

```
from tkinter import *

root = Tk()
root.title("Checkbox")
root.geometry("400x300")
state = IntVar()
cb = Checkbutton(root, text="Check me", onvalue=1,
                 offvalue=0, variable=state)
cb.pack()

if __name__ == "__main__":
    root.mainloop()
```



Radio Button

- Radio button je widget koji je najčešće deo grupe opcija od kojih korisnik može izabrati jednu.
- Sa argumentom ***variable*** elementi dugmadi se grupišu u određenu grupu. Variable može biti ***IntVar*** ili ***StringVar***.
- Sa argumentom ***value*** definiše se vrednost koju radio dugme ima.

Radio Button

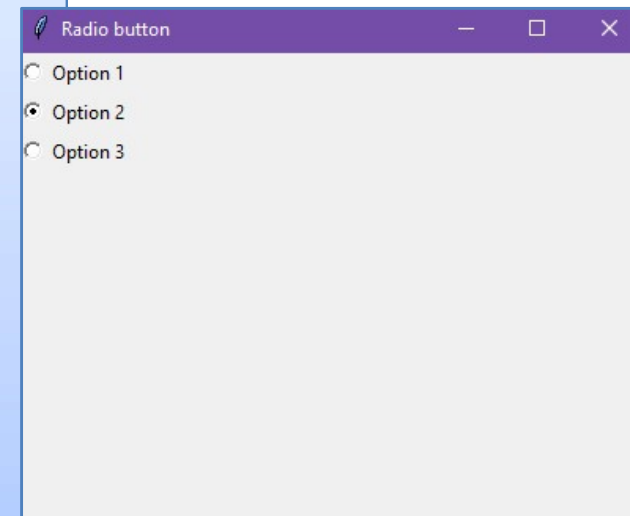
```
from tkinter import *

root = Tk()
root.title("Radio button")
root.geometry("400x300")
rbvalue = IntVar()
rbvalue.set(2)

rb1 = Radiobutton(root, variable=rbvalue, value=1, text="Option 1")
rb2 = Radiobutton(root, variable=rbvalue, value=2, text="Option 2")
rb3 = Radiobutton(root, variable=rbvalue, value=3, text="Option 3")

rb1.grid(row=0, column=0)
rb2.grid(row=1, column=0)
rb3.grid(row=2, column=0)

if __name__ == "__main__":
    root.mainloop()
```

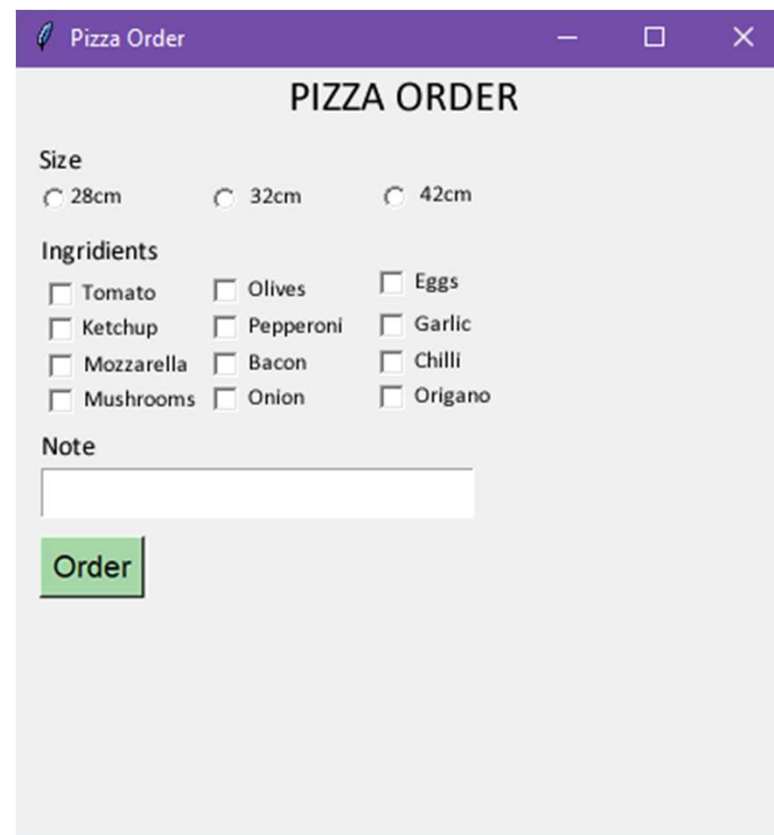


Primer

Kreirati interfejs kao na slici:

- **PIZZA ORDER**
- **Size** -> 28cm, 32cm, 42cm
- **Ingridients** -> Tomato, Olives, Eggs, Ketchup, Pepperoni, Garlic, Mozzarella, Bacon, Chilli, Mushrooms, Onion, Origano
- **Note**
- **Order**

Na klik dugmeta “Order” potrebno je kreirati narudžbinu i snimiti je u **timestamp.txt** gde je **timestamp** trenutni datum i vreme u terminu porudžbine (npr. **2021_05_12_2132.txt**)



The screenshot shows a web browser window titled "Pizza Order". The form is titled "PIZZA ORDER" and contains the following elements:

- Size:** Three radio buttons for "28cm", "32cm", and "42cm".
- Ingridients:** A grid of checkboxes for various toppings: Tomato, Olives, Eggs, Ketchup, Pepperoni, Garlic, Mozzarella, Bacon, Chilli, Mushrooms, Onion, and Origano.
- Note:** A text input field for additional notes.
- Order:** A green button to submit the order.

Literatura

- **Zvanična dokumentacija:**
<https://docs.python.org/3/library/tkinter.html>
- **Geeks For Geeks Tutorial:**
<https://www.geeksforgeeks.org/python-tkinter-tutorial/?ref=lbp>
- **Python StringVar tutorial:**
<https://www.pythontutorial.net/tkinter/tkinter-stringvar/>
- **Python Course Tkinter tutorial:**
https://www.python-course.eu/python_tkinter.php